מדריך קצר
למשתמשי UNIX

אילנה קפלן

אוניברסיטת בן גוריון בנגב

מרכז החישובים

תמוז תשנ"ב - יולי 1992

מדריך קצר
למשתמשי UNIX

אילנה קפלן

אוניברסיטת בן גוריון בנגב

מרכז החישובים

תמוז תשנ"ב - יולי 1992

אוניברסיטת בן גוריון בנגב

מערך המיחשוב

מדור תמיכה

משתמש יקר,

המדריך הקצר למשתמשי UNIX מוצא לאור ומופץ חינם באדיבות מרכז החישובים. במדריך זה נעשו מאמצים רבים ע״מ לענות על הצרכים הנובעים משימוש ב-UNIX. לשיפור השירות ולמתן אינפורמציה, אבקשך למלא את הספח ולשלוח אל אתי מלר במרכז החישובים או להכניס לתיבת השירות הנמצאת במדור.

☐ **מעוניין בקבלת מידע בנושאים הבאים ב-UNIX:** ______________ ______________

______________ ______________

☐ **מעוניין בקבלת מידע נוסף בנושאים הבאים:** ______________ ______________

______________ ______________

☐ **מעוניין בקבלת מידע מעמיק בנושאים הבאים:** ______________ ______________

______________ ______________

☐ **הצעות לשיפור העברת המידע:** ______________________________

שם פרטי ומשפחה ______________ טלפון ______________

פקולטה/מחלקה ______________ כתובת ב-E.MAIL ______________

שם פרטי ומשפחה ______ טלפון ______
פקולטה/מחלקה ______ כתובת ב-E.MAIL ______

תוכן העניינים

מ ב ו א

לשירות משתמשי מרכז החישובים קיימים כיום מספר תחנות עבודה המריצות מערכות UNIX והן:

masada	-	תחנת עבודה של RISC/6000 IBM במערכות הפעלה AIX .
arad	-	תחנת עבודה של RISC/6000 IBM במערכות הפעלה AIX.
chen	-	תחנת עבודה של DIGITAL במערכת הפעלה ULTRIX.
matar	-	תחנת עבודה של SUN במערכת הפעלה SUNOS 4.1.
ramon	-	תחנת עבודה של SYLICON GRAPHICS במערכת הפעלה IRIX.

המדריך מנסה להציג מושגים סטנדרטיים המתאימים לכל מערכות ה- UNIX. למרות הדמיון הרב ייתכנו הבדלים קטנים בין המערכות. במידת הצורך אפשר להתייעץ באנשי מרכז החישובים :

1. שמואל גרובר - טל. 461158 טל. פנימי 8158 .
2. מרינה מרקוס, או חיים לייב - טל. 461745 טל. פנימי 8745 .
3. אילנה קפלן, או ערן לקס - טל. 461160 טל. פנימי 8160 .
4. אלה אקרמן - טל. 461155 טל. פנימי 8155 .

חשבון, צופן וכניסה למערכת

כל אחת מתחנות ה-UNIX נגישות דרך BRIDGE .

כאשר מפעילים מסוף, תופיע בקשה : REQUEST

בתגובה לבקשה זו יש להקיש: BRIDGE

מהתפריט המוצג יש לבחור באפשרות הנושאת את המספר 15 ובהמשך יש להקיש: ' שם תחנה C '

לדוגמא : אם מעוניינים להתחבר ל- CHEN , נקיש: C CHEN .

כתגובה נקבל : login

בשלב זה יש להקיש את החשבון ובהמשך את הסיסמא של המשתמש. ניתן לקבל חשבון ע"י פניה לאיש הקשר המחלקתי או לאנשי החשבונות - ליאופולד או ישראל גלר ב-טל. 8743 .

החשבון שייפתח הוא חשבון ל- domain. ב- domain כיום נמצאים כל תחנות העבודה של מערך המיחשוב, לכן, החשבון והצופן שהמשתמש מקבל, מאפשר כניסה לכל אחת ממכונות ה- UNIX שהוזכרו במבוא.

קבצי המשתמש נמצאים פיזית באחת המכונות בלבד. למשתמש נראה שכל קבציו נמצאים בכל אחת מהמערכות, מכיון שהקבצים נגישים מכל מכונה ומערכת. מומלץ למשתמש העובד במערכות שונות, לשמור קבצים תחת directories שונים שיעזרו לו להבדיל בין המחשבים השונים.

בניגוד למערכות אחרות, UNIX מבדיל בין אותיות קטנות וגדולות. חשבון המשתמש הוא תמיד **באותיות קטנות** ולכן יש להקישן **באותיות קטנות**. הצופן מכיל לפחות 6 תווים. מומלץ לשנות את הצופן מידי פעם. הפקודה לשינוי הצופן היא :

yppasswd

כאשר מתקתקים את הצופן החדש, תהיה בקשה להקיש אותו שוב לשם אישור השינוי.

בסיום תהליך ה- login יופיעו על המסך הודעות מערכת ונוצר shell . ה - shell הוא 'command interpreter' הקולט את פקודות המשתמש והופך אותם לפקודות מערכת ההפעלה (ה- kernel) . ה- promt המופיע הוא מהצורה :

-מסי-/users/chen/cc/segel/david

שזה למעשה ציון ה -' home directory ' של המשתמש (הסבר יופיע בהמשך).

כדי לשנות את ה-promt למשהו אחר , (נניח לתו 'x') יש להקיש: set prompt = 'x'

בגמר העבודה יש לסיים את הקשר עם המכונה וזה נעשה באמצעות הפקודה: logout

במערכות מסויימות ניתן לסיים את העבודה ע"י הקשת ctrl d בו זמנית. (ctrl מציין לחיצה על המקש ctrl) .

Shell

מערכות ה- UNIX בנויות משני חלקים גדולים: kernel - לב ליבה של המערכת, המכיל את פקודות המערכת וה- shell שהיא command interpreter. היתרון במבנה זה שכל אחד יכול לבנות לו סביבת עבודה shell לפי ראות עינו מבלי לשנות את לב המערכת ה kernell.

ישנם כמה shell - ים :

sh - Bourne shell שפותח ב- AT&T .
ksh - Korn shell גירסה משופרת של ה- Bourne shell.
csh - שפותח בברקלי .
tcsh -

ה- tcsh הינו ה- shell המתקדם והנוח ביותר לעבודה אינטראקטיבית. היתרונות הם :

- כאשר עובדים תחת ה- tcsh , אפשר לחזור על פקודות קודמות ע"י הקשות חוזרות על המקש '⇌'
- תחת ה- tcsh אפשר להקיש ctrl d. במקום שם של קובץ, תוצג רשימה של הקבצים ב- directory הנוכחי והפקודה שתוקתקה מקודם. בשלב זה המשתמש יכול לבחור בנוחות את שם הקובץ הרצוי להפעלת הפקודה.
- הקשה על התווים הראשונים של הפקודה ולאחר מכן ctrl d , יביא להצגת רשימה של כל הפקודות המתחילות באותם תווים והצגה חוזרת של התווים הראשונים, וזאת כדי שהמשתמש יתקתק את המשך הפקודה הרצויה ויפעילה.
- ctrl u - מוחק את השורה שתוקתקה.
- ctrl w - מוחק מלה אחרונה.

- בהקשת הפקודה stty all נקבל טבלה המציגה ערכים הקשורים לעבודה עם המסוף. התו ' ^ ' המופיע בטבלה מסמן את המקש 'ctrl' .

פקודות UNIX

פקודות UNIX הן למעשה תוכניות. הקשת שם פקודה הינה בקשה להפעלת תוכנית מסויימת המאוחסנת במקום מסוים. התבנית הכללית של הפקודות היא : (פרמטרים אופציונליים) (אותיות אופציונליות) < שם פקודה >
שם פקודה יכול להיות שם באנגלית כמו date ויכול להיות צרוף אותיות לפי עקרון מסוים, כמו הפקודה grep הגזורה מ- ' global regular expresion printer '. הפקודה ls הגזורה מהמלה list או הפקודה cd הגזורה מהמלים change working directory.
אותיות אופציונליות או flags , הם תווים המשפיעים על אופן ביצוע הפקודה. בדרך כלל מקדים אותם התו ' - '. כאשר מציינים יותר מ-flag אחד מספיק לסמן ' - ' אחד לפני רצף התוים. למשל R- a- l- זהה ל- laR- .
לרוב מופיעים ה-flags לפני הפרמטרים.

דוגמאות לפקודות:

date
cp file1 file2
ls - al

אפשר לשים מספר פקודות בשורה אחת כאשר התו ' ; ' מפריד בין הפקודות.
כאשר פקודה מסוימת קיימת במערכת במקומות שונים , נניח משתמש כתב תוכנית באותו שם כמו פקודת מערכת, אפשר לבדוק איזה פקודה מתבצעת ע"י הפקודה :
which פקודה

כתגובה נקבל את המסלול המלא של הפקודה (התוכנית) והיא זו המופעלת.
ניתן להפסיק פקודה ע"י ctrl c .

PIPING, REDIRECTION

פקודות unix שהם למעשה תוכניות, הינם בעלי קלט ופלט. לרוב, הפקודות מצפות לקלט מהמקלדת (הנקרא 'the standart input') ולפלט למסך (הנקרא 'the standart output'). יש ולפעמים נוח יותר לקרוא את הקלט מקובץ ולהדפיס תוצאות על קובץ מבלי לשנות את התוכנית. השינוי באופן קליטת הקלט והפלט נקרא redirection.
redirection של הקלט נעשה ע"י הסימן ' > '.
כדי לעשות redirection של הפלט לקובץ - נשתמש בסימן ' < '.
כדי שהפלט יתוסף לקובץ קיים ולא יעלה על הקיים, נשתמש בסימן ' << '.
לדוגמא: כאשר רוצים לשמור על קובץ את רשימת המשתמשים במערכת ברגע זה, להוסיף אל הקובץ את התאריך, ולשלוח את הקובץ בדואר, נפעיל את הפקודות הבאות:

who > usersfile
date>>usersfile
mail xxxx < usersfile

הרצת תוכנית עם פלט ארוך המהווה קלט לתוכנית אחרת, ניתן לביצוע בפקודה אחת ע"י שימוש בסימן ' | ' הנקרא piping או צינור.
לדוגמא : כדי לקבל רשימה ממוינת של המשתמשים העובדים כרגע , נפעיל את הפקודה who | sort
בכך גרמנו שבמקום שהפקודה who תדפיס את הפלט שלה על ה- standart output ה-pipe תופס את הפלט ומעביר אותו כקלט לפקודה sort. באופן כללי הפקודה :

קובץ$_3$ < תוכנית$_3$ | תוכנית$_2$ | קובץ$_1$ > תוכנית$_1$

מריצה תוכנית ראשונה הקולטת קלט מקובץ$_1$, תוכנית שניה העובדת על הפלט של התוכנית הראשונה והתוכנית השלישית העובדת על הפלט של התוכנית השניה ושומרת את הפלט שלה על קובץ$_3$.

STANDART ERROR

ברירת המחדל של מערכת ה- unix היא להציג הודעות שגיאה על המסך , כדי לקבל את הודעות השגיאה על קובץ, נשתמש בסימן ' >& ' (redirection &) לדוגמא :

```
cc f1 >& err
```

הפקודה מבצעת קומפילציה של הקובץ f1 תוך העברת הודעות שגיאה לקובץ err .

CONTROL CHARACTERS

ישנם כמה control charcters שאם הם נשלחים מהמסוף, יש להם משמעות.

- ctrl d - מבצע logout, ועובד כמו end - of - file בזמן הקשת נתונים מהמקלדת.
- ctrl c - מסיים את הפקודה שרצה כרגע.
- ctrl z - משהה (לא מסיים) את הפקודה הרצה כרגע.
- ctrl u - מוחק את השורה שתוקתקה ועדיין לא נכנסה לביצוע.
- ctrl s - עוצר את הדפסת הפלט על המסך (scrolling).
- ctrl g - ממשיך את הדפסת הפלט על המסך.

קבלת עזרה MAN

כל פקודה ב- unix ורוטינות ספריה רבות מתועדות, מפורטות ומוסברות ב- 'manual pages' המהווה 'online manual' למשתמש.

ה- 'manual pages' מחולק ל -8 פרקים :

1 - פקודות לשימוש כללי
2 - פקודות מערכת (עבור מתכנתי UNIX)
3 - רוטינות ספריה (למתכנתים)
4 - אמצעי I/O
5 - קבצי מערכת
6 - משחקים
7 - חבילות Macro
8 - כלים למתכנת מערכות

הפקודה לקבלת ה- 'manual pages' על המסך היא: שם פקודה man

למשל , כדי לקרוא על הפקודה man יש להקיש: man man

אם הפלט ארוך , הפקודה man עוצרת אחרי כל מסך ויש להקיש על מקש הרווח כדי לקבל את המסך הבא. בסוף ה- 'manual pages' יש קטע עם הכותרת See Also המציין פקודות נוספות שכדאי לקרוא.

כדי להשתמש ב-man עלינו לדעת את שם הפקודה . במידה ואנו לא יודעים את שם הפקודה נשתמש בפקודה man -k keyword. כאשר ה-keyword הוא מלה שתרמוז על הפקודה. במדה ומלה זו תופיע בכותרת ב-'manual pages' של פקודה מסויימת, אז נקבל שורה בפלט המציינת את שם הפקודה, באיזה פרק מפרקי ה- man היא מתועדת (המספר המופיע בסוגריים) והכותרת לאותה פקודה. לדוגמא: כדי למצוא את הפקודה להעתקת קבצים , נשתמש ב-keyword 'copy' ונקיש את הפקודה:

man - k copy

אחת משורות הפלט תהיה:

cp(1) - copy file data

שהיא הפקודה להעתקת קבצים.

לקבלת פירוט מלא על הפקודה cp נשתמש כעת בפקודה: man cp

כאשר אותה פקודה מופיעה במספר פרקים, יש לציין את מספר הפרק בפקודה man , אחרת הפקודה man תציג את ההסבר על המופע הראשון של הפקודה, כפי שהיא תופיע ב-'manual pages' .

man משתמש בסימונים קבועים בהצגת הפקודה. הסימנים הם :

- [ ] - המסמן שהערך המופיע בתוך הסוגריים הוא אופציונלי.
- שם או { } - המסמן שהערך בסוגריים הינו חובה.
- | - המסמן בחירה בערך אחד בלבד. למשל [a | b] מסמן שאפשר לבחור או a או b .
- ... - המסמן שהפרמטר האחרון המופיע לפני שלוש הנקודות יכול לחזור מספר פעמים.
- ____ - המסמן standart input .

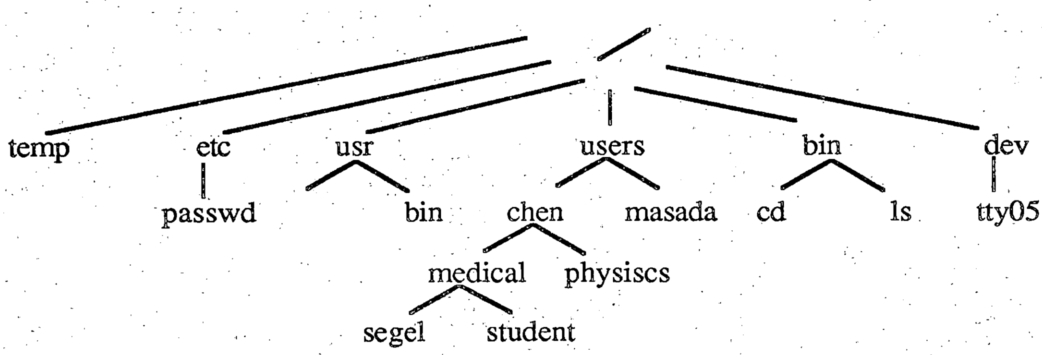
כאשר ידוע שם הפקודה וברצוננו לקבל הסבר קצר על הפקודה יש להקיש :

whatis פקודה

ובכך נקבל את הכותרת של הפקודה כפי שהיא מופיעה ב- 'manual pages' .

קבצים ו-DIRECTORIES

הקבצים ב- unix מאורגנים במבנה הנקרא 'file system' . המבנה הנ"ל הוא היררכי והקבצים מאוחסנים תחת directories ואלו מאוחדים בעוד directories ובראש עומד directory אחד הנקרא root המסומן במערכת ע"י הסימן ' / ' . מערכת קבצים סטנדרטים ב- unix נראית בערך כך :



העץ הנ"ל הוא חלקי

כאשר :
- / bin מכיל פקודות של מערכת ההפעלה (בד"כ קבצים בינאריים)
- / etc מכיל קבצי ניהול .
- / lib מכיל ספריות.
- / tmp מקום לקבצים זמניים. (בד"כ המערכת מוחקת בזמנים קצובים את תוכן ה- directories הנ"ל) .
- / dev מכיל קבצי device של המערכת.
- / users מתחתיו יושבים המשתמשים (home directories של המשתמשים) .

המושג "current directory" מוגדר כ- directory בו עובדים באותו רגע.
הפקודה לקבלתו היא : pwd

לכל משתמש יש גם "home directory" והוא ה- directory בבעלות המשתמש ושם הוא מחזיק את קבציו.
בזמן ה- login ה- 'current dir' זהה ל- 'home dir'.
לעבור מ- dir אחד לאחר יש להקיש את הפקודה : cd

למשל : cd /bin - יעביר את המשתמש ל- bin dir /.
cd - בלי פרמטר יחזיר את המשתמש ל- 'home dir'.
.. - הינו סימון ל 'parent directory' זהו ה- directory שעבורו ה-current directory הוא sub directory ראשוני.
. - הינו סימון ל - current directory.
~ - הינו סימון ל - home directory של המשתמש.
~ xxx - הינו סימון ל- home directory של המשתמש xxx
ls - הינה הפקודה לקבלת רשימת הקבצים ב- directory. (הסבר על הפקודה מופיע בהמשך המדריך).

ישנם שלושה סוגי קבצים ב- unix :

1. קבצים רגילים המכילים נתונים, תוכניות וכו'.
תת קבוצה של קבצים רגילים הנקראים "hidden files", אלו קבצים שלא מופיעים בתצוגה הפשוטה של תוכן ה- directory. קבצים אלו מאופיינים ע"י ' . ' בתחילת שם הקובץ והם לרוב קבצי מערכת, קבצים שהמערכת מצפה למצוא אותם בשמות directorys קבועים.
2. "directories files" הוא קובץ שמכיל בתוכו מידע על קבצים אחרים, כאילו אינדקס לקבצים שיש תחתיו directory ב- unix הינו קובץ.
3. special files - קבצים לניהול המערכת, בקרה על מסופים, devices שהמשתמש אינו צריך לגעת בהם לרוב.

שם של קובץ יכול להיות בנוי מכל תו אלפביתי, ספרות+אותיות והסימנים . , _ + - כאשר יש הבדל בין אותיות קטנות וגדולות. השם יכול להגיע עד 255 תווים.
יש סיומות מחייבים לקבצים מסויימים, למשל :
- סיומת f. חובה לקובץ המכיל תוכנית פורטרן.
- סיומת c. חובה לקובץ המכיל תוכנות ב- c.
- סיומת p. חובה לקובץ המכיל תוכנות ב- פסקל p.

/ dev / null מסמן ריק, קובץ שעובר לשם נעלם ואם נקרא ממנו נקבל שהוא ריק.
כל קבצי ה- unix הינם במבנה אחד ואין לקבצי unix מבנה פנימי כמו index או רשומות.
שם מלא של קובץ מכיל את המסלול מה- root אל הקובץ כולל כל ה- directory עד לקובץ. המבדיל בין המרכיבים במסלול הוא התו ' / ' (זה אותו תו המסמן גם את ה- root).

לדוגמא : /users/chen/cc/segel/david/result

אין צורך לציין את המסלול המלא ואפשר תמיד לציין את המסלול היחסי ל- current dir.
למשל אם ה-current dir הוא /usr/bin אז spool /.. ו- /usr/spool הם זהים.

WILDCARD

ה- shell נותן משמעות לסימנים מסויימים המופיעים בתוך שם של קובץ ובכך הם יקצרו את מה שעלינו לתקתק.

התו '*' - מסמן אפס או יותר תווים (תו כלשהו) למשל, a*b יתאים ל- ab , ayfb , a327b כלומר, שמות המתחילים ב- a ומסתיימים ב- b .

התו '?' - מסמן תו אחד כלשהו למשל a ? b מתאים ל- a 1 b אבל לא ל-ab

[gw 0-9] - מתאים לכל תו מהקבוצה g , w או מהתחום מ- 0 עד 9 למשל ab[xy 2-5] c יתאים ל- abxc ל- ab3c אבל לא ל- abdc .

דוגמא:

כדי לקבל הדפסה של כל קבצי הפורטרן של המשתמש בשם kuku נתקתק :

ls kuku/*.f

יצירה ומחיקת DIRECTORIES

כדי ליצור directory יש להשתמש בפקודה : mkdir שם

כדי למחוק directory יש להשתמש בפקודה : rmdir שם

rmdir תמחוק את ה- directory רק אם הוא ריק.

directory או קובץ שנוצר לא יורש תכונות מה- directory בו הוא נוצר. התכונות נרכשות מה- user שיצר אותם.

הגנה על קבצים

ישנם שלושה סוגי הגנה לקובץ :

- הגנה מפני בעל הקובץ, u (user).
- הגנה מפני ה- group אליו משתייכים, (g) . group הוא מושג ניהולי, כל המשתמשים מחולקים לקבוצות, בדרך כלל לפי המחלקות אליהם הם שייכים .
- הגנה מפני אחרים o (others) . other הם כל המשתמשים פרט ל- group .

בכל סוג הגנה ישנם שלוש דרגות הגנה. הגנה מפני קריאה r (ready) , הגנה מפני כתיבה w (writing) והגנה מפני ביצוע x (execute) . ב- unix ההגנות על הקבצים נקראים modes . כדי לקבל מה ההגנות של הקבצים משתמשים בפקודה :

ls -1

וכתגובה מקבלים רשימה שתראה דומה לשורה הבאה :

-rwxr-xr-x 1 focus 237 Nov 27 12:35 results

כאשר :

- התו ' - ' בתחילת השורה מציין שמדובר בשם קובץ. d במקום זה מציין שם directory .
- rwxr-xr-x מציין את ה- mode של הקובץ. שלושת התווים הראשונים ב- mode מציינים את ההרשאות לבעל הקובץ. שלושת התווים הבאים מציינים הרשאות של ה- group והשלושה האחרונים מציינים את הרשאות לאחרים. w מציין הרשאה לכתיבה. r מציין הרשאה לקריאה, x מציין הרשאה לביצוע. התו '-' מציין שאין הרשאה.
- 1 מציין את מס׳ הקשרים שיש לקובץ זה לקבצים אחרים. בדוגמה זו אין קבצים **קשורים** לקובץ הזה פרט לקובץ עצמו.
- 237 מציין את גודל הקובץ בתווים.
- focus הוא שם בעל הקובץ.
- Nov 27 12:35 הוא התאריך והזמן לשינוי האחרון שנעשה בקובץ.
- results הוא שם הקובץ.

אפשר לשנות mode של קובץ ע״י הפקודה chmod , יש לציין למי נשנה הרשאה (user, group או other) האם נרצה לאשר (+), או האם נרצה לבטל (-) הרשאה, סוג ההרשאה x,w,r ולבסוף שם הקובץ : לדוגמא -

chmod g+w , o-r filename

ברירת המחדל של הקבצים היא rw - r - - r - - אם נרצה לבטל זאת נוכל להפעיל את הפקודה :

chmod g - r , o - r *

ובכך נוריד מכל הקבצים שברשותנו את אופציות ה- read ל- group ול- others .

פעולות בסיסיות על קבצים

קבלת רשימת קבצים - ls

ls [flags] [directory]

בלי שום פרמטר מקבלים רשימת קבצים ב- directory הנוכחי.

flags שימושיים הם :

- l - לקבלת פלט בפורמט ארוך (long) בו תוצג שורה לכל קובץ כולל גודל בתווים, זמן השנוי האחרון, modes וכו׳.
- a - לקבלת פירוט על קבצים שהשמות שלהם מתחילים בתו ' . ' .
- R - לקבלת רשימת קבצים גם של subdirectories (ביצוע ls בצורה רקורסיבית).
- F - לקבלת פלט בפורמט מקוצר בו תתווסף הסיומת ' * ' לקובץ executable ו- ' / ' לשמות directories.

העתקת קבצים - cp

cp [פרמטרים אופציונליים] file1 file2

cp [פרמטרים אופציונליים] file ... directory

cp [פרמטרים אופציונליים] directory ... directory

cp מעתיקה קובץ$_1$ לקובץ$_2$, ההרשאות והבעלות על קובץ$_2$ נשמרים אם קובץ זה כבר קיים אחרת ההרשאות והבעלות של קובץ$_1$ יועברו לקובץ$_2$.

בצורה השניה קובץ אחד או יותר יועתקו ל - directory עם שמם המקורי.

בצורה השלישית directory אחד או יותר יועתקו ל- directory חדש.

בין הפרמטרים האופציונלים נציין ש :

i - inquire יגרום להופעת שאלה למשתמש כאשר יש מצב שפקודת ההעתקה תעלה על קובץ קיים. תשובה yes תגרום להמשך ההעתקה ועליה על הקובץ הקודם.

r - יעתיק את ה- directory עם כל ה- subdirectory והקבצים שבכל אחד מהם.

<u>העברה או שינוי שם של קובץ - mv</u>

mv [פרמטרים אופציונליים] <u>file1</u> <u>file2</u>

mv [פרמטרים אופציונליים] <u>file1</u> ... <u>directory</u>

במקרה הראשון שם הקובץ משתנה מ- file1 ל- file2 והקובץ עובר מ- file1 ל- file2 ו- file1 נעלם.

במקרה השני קובץ אחד או יותר או directory מועברים ל- directory שני. בין הפרמטרים האופציונליים ישנו i - הפועל באופן זהה ל-i - בפקודה cp.

<u>מחיקת קובץ - rm</u>

rm [פרמטרים אופציונליים] file או directory ...

מוחק קובץ אחד או יותר.

פרמטריים אופציונליים הם :

r - מחיקה רקורסיבית של כל ה- subdirectories ותוכנו.

i - לבקשת אישור לפני כל מחיקה.

f - לבצע מחיקה בכוח (force) גם עם הקובץ ' read only '.

<u>להציג קובץ על המסך - tail, head, more</u>

more קובץ

התצוגה תעצר בסוף כל מסך . כדי להמשיך למסך הבא יש להקיש רווח. כדי לקבל את השורה הבאה יש להקיש על המקש return.

הצגת כמה שורות מתחילת הקובץ נעשית ע"י הפקודה:

head קובץ

הצגת כמה שורות מסוף הקובץ נעשית ע"י הפקודה:

tail קובץ

חיבור והצגת קובץ - cat

cat [פרמטרים אופציונליים] file ...

הפקודה cat קוראת את הקבצים לפי הסדר ומציגה אותם על הפלט הסטנדרטי , לכן, כדי להציג קובץ על מסך נפעיל את הפקודה :

cat שם קובץ

כדי לחבר שני קבצים וליצור קובץ שלישי נפעיל

cat $_1$קובץ $_2$קובץ < $_3$קובץ

אם לא נציין קובץ אזי - cat יקליט מהקלט הסטנדרטי (מהמקלדת) את הנתונים. לכן כדי ליצור קובץ נפעיל :

cat < קובץ

נסיים את הכנסת הנתונים ע"י ctrl d המסמן end of file.

תפיסת הדיסקים

כדי לדעת מה השטח הפנוי ב- file system יש להשתמש בפקודה :

df

הטבלה המתקבלת מציגה את תפיסת הדיסקים ב- kbytes .כאשר השטח הכללי מורכב מסכום השטח המנוצל, השטח החופשי והשטח השמור (בערך כ- 10%).

כדי לדעת את תפיסת הדיסקים של קבצי המשתמש בלבד יש לתקתק

du

ציון הפרמטר האופציונלי s - תגרום לתצוגת ס"כ התפוסה של המשתמש. ואם פרמטר a - נקבל את הגודל של כל קובץ . הגדלים הם ב- kbytes .

מושגים הקשורים לקבצים

/tmp

אם למשתמש קובץ גדול שהוא מעונין לשמור למעט זמן ואיננו מעונין שהקובץ יגובה בתהליך הגיבוי של המערכת אזי אפשר לאכסן קובץ זה ב- tmp directory/ .

tmp/ הינו שטח ציבורי פתוח לכל משתמש. הקבצים במקום זה נמחקים בלי הודעה מוקדמת. המחיקה נעשית בדרך כלל כאשר השטח המיועד ל- tmp/ מפוצץ ואין בו יותר מקום.

compress

כאשר למשתמש קובץ גדול שיש לשומרו לאורך זמן ויש צורך לחסוך במקום, אפשר להפעיל על הקובץ את פקודת הדחיסה

compress filename

בכך אפשר לחסוך עד 50% מהמקום הראשוני. הקובץ שיבצר יהיה בשם "filename . Z" כלומר מתווספת סיומת Z . והקובץ המקורי נעלם. כדי להחזיר את הקובץ למצבו הרגיל מפעילים את הפקודה

uncompress filename .Z

הקובץ עם הסיומת Z . יעלם ובמקומו יופיע הקובץ filename כמו בתחילת התהליך.

גיבוי ושחזור

קבצי המשתמש מגובים במערכות ה- UNIX מידי יום. הקלטות המיועדות לכך מספיקות לתקופה מוגבלת , כאשר מטרת מרכז החישובים להגדיל בעתיד תקופה זו, עד שלושה חודשים. במידה ובטעות נמחק קובץ, אפשר לבקש לשחזרו מאנשי ה-system במרכז החישובים.

משתמש שעוזב את המערכת לתקופה מסוימת מומלץ שיגבה את קבציו לקלטת לשם שמירה.

הדפסה

הפקודה לשליחת קובץ להדפסה היא lpr בצירוף שם המדפסת. כרגע הדפסות עם unix מתבצעות דרך NOS/VE ו- VMS לכן, אם יש עיכוב בקבלת פלט יש לבדוק את מצב ה- NOS/VE וה- VMS .

פורמט הפקודה הוא:

lpr - Pשם מדפסת filename

כאשר שם המדפסת יכול להיות :

lpz - למדפסת הלייזר במרכז החישובים

lps - לקובץ postscript במדפסת הלייזר

lpcenter - מדפסת מרכזית במרכז החישובים

lp - המדפסת בבית היאס דרך BGUVMS

ניתן לראות את מצב תור ההדפסה של כל מדפסת ע"י:

lpq - Pשם מדפסת

שירותי Shell

alias

alias הוא פקודה המאפשרת לתת שם קצר לפקודה מסובכת או לפקודה המופעלת עם הרבה פרמטרים לדוגמא :

alias ls 'ls - l'

תגרום לכך שכל פעם שנכניס את הפקודה ls תתבצע הפקודה 'ls - l' .

הפקודה alias ללא ציון פרמטרים מציגה רשימה של כל ה- alias שה- shell מכיר באותו רגע.

history

ה- shell שומר רשימה של כל הפקודות שהוכנסו. כדי לקבל את רשימת הפקודות, יש להפעיל את הפקודה :

history

כתגובה נקבל את רשימת הפקודות שבוצעו, זמן ביצוע הפקודה ומספרה הסידורי ברשימה.

ניתן להגדיר את גודל הרשימה שתשמר. כדי לשמור רשימה של עד n פקודות נפעיל את הפקודה :

set history = n

redo

- `-` — כל לחיצה על-״ – ״ תציג פקודה אחת אחורה.
- `מסי !` — חוזר על פקודה מסי
- `!!` — תפעיל את הפקודה האחרונה שוב.
- `מחרוזת !` — יפעיל את הפקודה הכי קרובה בזמן שהתחילה במחרוזת המצוינת.

כדי להפעיל פקודה קודמת בשינוי מה יש לתקתק:

`p : מסי !` — שתגרום לכך שפקודה מסי תוצג על המסך. על פני שורת הפקודה יש לזוז עם החיצים " . " ",", כל תו שנקיש נכנס ב-insertmode. כדי למחוק תו יש לעבור אותו ולהקיש על המקש המסומן ב- ⌫ x . ctrl e מקפיץ את הסמן לסוף הפקודה ו- ctrl a מקפיץ את הסמן להתחלת הפקודה. כדי לשנות בפקודה משהו יש להפעיל מחרוזת חדשה ^ מחרוזת ישנה ^ . עם לחיצה על מקש Return, הפקודה תתבצע.

לדוגמא :

history	
23 rm table.h	
! 23:p	
rm table.h	יופיע:
^table^tbl	נקיש:
rm tbl.h	יופיע:

והפקודה תתבצע.

אם הפקודה לביצוע היא הפקודה האחרונה, אנו מבצעים את השינוי בלבד ואין צורך להציג את הפקודה קודם.

path

פקודות unix הם תוכניות המאוכסנות ב- directories כמו bin / או usr / bin / .
במשך הזמן מתקינים תוכנות ופקודות חדשות ב- directories שונים. וכאשר מתקתקים פקודה ה- shell סורק מסי directories בהם הוא מצפה למצוא את הפקודה, רשימת ה- directorie לחיפוש נמצאת במשתנה path . כדי לדעת מה הרשימה הנוכחית יש להפעיל את הפקודה set ולשים לב למשתנה path. כדי להוסיף directories ל- path הקיים, יש להפעיל את הפקודה:

set path = ($ path directory$_1$ directory$_n$)

כדי לדעת היכן פקודה מסוימת מאוכסנת נפעיל את הפקודה

whereis פקודה

וכתגובה נקבל את המסלול המלא של הקובץ, המכיל את הפקודה ואז אפשר או להוסיף את ה-directory ל-path או להפעיל את הפקודה עם ה-path המלא.

הקובץ login. cshrc.

הקובץ login. מכיל פקודות shell המתבצעות בזמן הכניסה (login) למערכת. הפקודות כמו path, history , alias משפיעות על ה- shell הנוכחי ולכן במקום לתקתק אותם כל פעם מכניסים אותם לקובץ login .
יש פקודות שבהפעלתם נפתח shell חדש. ה- shell הנ״ל לא יבצע את ה- login . במידה ויש פקודות שצריכות להתבצע עם פתיחת כל shell , אז יש להכניס פקודות אלו לקובץ cshrc.
כל חשבון שנפתח מכיל קובץ login , ו- cshrc סטנדרטי . המשתמש יכול בכל רגע להוסיף לקבצים אלו פקודות.
כדי לקבל מידע נוסף על פקודות " c shell " יש להקיש :

man csh

עבודות - Jobs

מערכות UNIX סטנדרטיות אינן מכירות בעבודת Batch. המשתמש יכול להריץ יותר מ -job אחד ואלו יכולים להמשיך לעבוד גם אחרי ביצוע logout .
כאשר מתקתקים פקודה הפקודה מתבצעת ועל המשתמש להמתין עד גמר הפקודה ורק אז הוא יכול להמשיך להשתמש במסוף. דרך עבודה זו היא ' running in the foreground ' .
הדרך השניה היא ' running in the background ' במצב זה ה- job מתנתק מהמסוף וממשיך לעבוד ברקע ובמקביל המשתמש יכול להמשיך לעבוד במסוף. בדרך זו ה- job ימשיך לעבוד גם אחרי logout . ניתן להחליט בכל שלב בהתחלה או באמצע ההרצה על העברת ה- job ל- background .

כדי להריץ את ה- job מהתחלה ברקע יש להוסיף את התו ' & ' בסוף הפקודה.
לדוגמה: כדי להריץ תוכנית בשם simulater ברקע נתקתק :

```
simulater &
```

אם ה- job נמצא בעבודה ונרצה להעבירו לרקע, נתקתק בתחילה ctrl z וזה ינתק את ה- job מהמסוף וכאשר מתקבל ה- prompt נכתוב bg ובכך ה- job ימשיך ברקע.

כדי לבדוק את מצב העבודות נשתמש בפקודה:

```
jobs
```

ונקבל מידע על העבודות שהופסקו או שרצים. לדוגמא:

```
[1] + Running simulator
[2] - Stopped f77 prog, f
```

כדי להחזיר job מהרקע (מה-background) ל-foreground, נצטרך את מספר ה-job כפי שהוא מופיע בפלט של הפקודה jobs ואז להקיש את הפקודה:

```
fg % job number
```

לדוגמא:

```
fg % 2367
```

כדי להפסיק jobs ב-background יש להפעיל את הפקודה kill ומספר ה-job. לדוגמא:

```
kill % 10
```

process (תהליך)

מצד המשתמש אנו מריצים jobs, מצד המערכת כל פעולה מתבצעת ע״י process אחד או יותר.
כדי לראות את כל התהליכים העוברים כרגע תחת חשבון המשתמש כולל מ-shell אחר, נשתמש ברמת ה-process הפקודה לקבלת ה-process היא ps - (print process status statistic)

```
ps
```

הפלט המתקבל נראה כך:

PID	TT	STAT	TIME	COMMAND
1214	p8	S	0:00	-csh
2915	p8	R	0:00	ps
2194	pd	S	0:00	-csh(tcsh)
2050	pd	R	0:00	ps
1041	p1	R	9:21	simmulator
1032	p0	I	0:01	telnet bgunve

נקבל את הפרטים הבאים:

- process id המגדיר באופן יחיד את ה-process
- המסוף ממנו התחילו את התהליך
- status התהליך כאשר:

R	מסמן	running
T	מסמן	stopped
P	מסמן	page wait
D	מסמן	disk wait
S	מסמן	sleeping < 20 second
I	מסמן	idle > 20 second

- CPU time
- הפקודה שיצרה את התהליך.

בכל מצב שנפעיל את הפקודה ps נקבל את שני התהליכים הבאים: האחד הוא ה-ps והשני הוא ה-shell של התהליך הנוכחי והוא: csh- .

כדי לקבל פלט על כל התהליכים השיכים למשתמש 'kuku' אפשר להפעיל את הפקודה:

ps aux | grep kuku

הפרמטרים aux מאפשרים לקבל את רשימת כל התהליכים הקיימים כולל את אלו שממשיכים לעבוד אחרי log out

a - מסמן שיש לציין את כל התהליכים שמתבצעים מהמסוף

x - מסמן שיש לציין את כל התהליכים שלא ממסוף

u - מסמן לשיש להציג את המידע המיוחד ל-users.

כדי לסיים תהליך יש להשתמש בפקודה:

kill מסי תהליך

כאשר מסי תהליך הוא PID המופיע עם פלט הפקודה ps במידה וה-kill לא זרק את התהליך אפשר להפעיל את הפקודה:

kill - 9 מסי תהליך

או

kill - 9 % job מסי

האופציה 9 - מציינת שיש להפסיק את התהליך באופן סופי.

העברת קבצים - ftp

כדי להעביר קבצים בינאריים, או קבצים גדולים בין תחנות עבודה או מחשבים, יש אפשרות להשתמש ב-ftp.

יש לתקתק:

ftp [פרמטרים אופציונליים] שם מכונה

בהמשך תופיע בקשה להכנסת חשבון וצופן המשתמש במכונה המרוחקת.

כאשר הקשר נוצר, אפשר להקיש פקודות ftp ביניהם:

cd - לשנות directory במכונה המרוחקת
lcd - לשנות directory במכונה הנוכחית
pwd - הצגת שם directory במכונה המרוחקת
ls - הצגת רשימת קבצים במכונה המרוחקת
get שם קובץ - להביא קובץ מהמכונה המרוחקת למכונה הנוכחית
put שם קובץ - לשלוח קובץ מהמכונה הנוכחית למרוקחת
mget שמות קבצים - להביא מספר קבצים (יש אפשרות להשתמש ב-wildcard)
mput שמות קבצים - לשלוח יותר מקובץ אחד (יש אפשרות להשתמש ב-wildcard)
bin - לעבור ל-binary mode הכרחי להעברת קבצים בינאריים
ascii - חזרה ל-ascii mode (זוהי ברירת המחדל)
quit - יציאה מה-ftp

פרמטרים אופציונליים יכולים להיות:

i - מונעת הופעת בקשות לאישור במקרה של העברת מסי קבצים
v - גורם לתצוגת סטטיסטיקה ותגובות מהמכונה המרוחקת.

שפות תכנות וקומפילציות

בכל אחת מתחנות ה-unix אפשר לעבוד ב-FORTRAN.
הפקודה לקימפול תוכנית ב-Fortran שונה בתחנות השונות, ויש גם הבדלים בפרמטרים מסויימים.
ב-chen הפקודה היא:

f77 [פרמטרים אופציונליים] prog.f [פרמטרים אופציונליים]

ב-masada וב-arad

xlf [פרמטרים אופציונליים] prog.f

אם הקומפילציה עברה בשלום נוצר קובץ בשם a.out. כדי להריץ את התוכנית יש להקיש:

a.out

הפרמטרים האופציונליים הנפוצים יותר בזמן הקומפילציה הם:

c - לקמפל מבלי לבצע linking
o שם קובץ - ליצור קובץ בינארי בשם המצויין ולא בשם הסטנדרתי a.out
g - להכין קובץ בינארי מוכן לריצה עם debugger
O - לבצע אופטימיזציה
l <key> - לצרף את הספריה lib<key>.a לחיפוש רוטינות חסרות. לדוגמא:

f77 - g calc.f - 0 calc

יש לציין שהפורטרן עובד עם מסי 'logical units' קבועים:

0 - מיועד ל-'standart error'
5 - מיועד ל-'standart input'
6 - מיועד ל-'standart output'

להפעלת המחדר (compiler) לשפת C משתמשים בפקודה:

cc שם קובץ
xlc שם קובץ או ב- IBM/AIX

להפעלת מהדר ל-Pascal משתמשים בפקודה:

pc שם קובץ
xlp שם קובץ או ב- IBM/AIX

לקבלת הודעות השגיאה של הקובץ יש להשתמש בתו & כפי שהוסבר בסעיף standart error.
לדוגמא:

xlf f1.f > & error file

הודעות השגיאה יכתבו לקובץ error.file.

פקודות שימושיות נוספות

חיפוש ביטוי בקובץ - grep

grep - Global regular expression printer הינה תוכנה המאפשרת לקבל כפלט את כל השורות בהם מופיע ביטוי מסויים.

הפקודה היא:

grep [פרמטרים אופציונליים] ביטוי [קובץ . . .]

כאשר חלק מהפרמטרים האופציונליים הם:

m - לקבלת ציון מספר שורה בתצוגה

c - לקבלת שם קובץ ומספר המציין כמה פעמים נמצא ביטוי מצויין בקובץ

l - לחיפוש ביטוי ב-directory מסויים תתקבל רשימה של שמות הקבצים והשורות בהם מופיע הביטוי.

y - לא להבדיל בין אותיות קטנות לגדולות

v - להצגת כל השורות שלא מכילים את הביטוי המצויין

הביטוי שאנו מחפשים יכול להכיל סימנים בעלי משמעות והם:

'^' - סימון תחילת שורה. לדוגמא: grep '^Sally' יחפש את כל השורות המתחילות במילה Sally

'$' - סימון סוף שורה. לדוגמא: grep 'Smith$' יאתר את כל השורות המסתיימות במילה Smith

'^$' - סימון לשורה ריקה

'.' - מסמן תו אחד כלשהו

[קבוצה] - מסמן שכל תו מהתוים הנמצאים בקבוצה יכולים להופיע

[a - z] - התו "-" מסמן רצף [a - z] שקול ל-[abcd......z]

'*' - מסמן תו כלשהו אפס או יותר פעמים

'+' - מסמן תו כלשהו שיופיע פעם אחת או יותר

'?' - מסמן תו כלשהו שיופיע אפס או פעם אחת בלבד

אם אחד מהתוים הנ"ל צריך להיות חלק מהביטוי ולא לשמש כסימן בעל משמעות מורחבת לפקודה grep יש לשים ' \ ' לפני התו.

דוגמאות:

grep "Sally Smith" file

יציג את כל השורות מוקבץ file המכילים את השם Sally Smith.

במקום file יכול להיות גם רשימה של קבצים ואז בפלט יופיע גם שם הקובץ שבו השורה מופיעה.

grep - l kuku * p

יציג את רשימת כל הקבצים ששמם מסתיים ב-p והם מכילים שורה המכילה את הביטוי kuku.

SOURCE

כאשר מבצעים מס' פקודות מס' פעמים אפשר ליצור קובץ עם הפקודות החוזרות ולהפעילם ע"י:

source file

עבודה עם TEX

משתמשים יכולים להריץ קבצי input ל-tex.שם הקובץ חייב להסתיים עם ".tex" .
אפשר להריץ כרגע ב- chen (מכונת ה-unix של digital) tex, latex ו-slitex לשם כך יש לתקתק:

tex xx.
latex xx.
slitex xx.

כאשר ב-directory נמצא קובץ xx.tex הקובץ המתקבל הוא xx.dvi (device independent dvi), כמו כן מתקבל קובץ xx.log

כאשר ישנם הודעות שגיאה מתקבל prompt של tex המחכה לקלט.
האפשרויות הם:

(cr) - להמשיך
s - Scroll לטעות הבאה
r - לרוץ בלי לעצור
q - לרוץ בשקט
i - להכניס שינוי
e - לערוך את הקובץ
h - לקבלת help
x - ליציאה quit

כדי לקבל תצוגה של הקובץ על המסך יש להשתמש בפקודה:

xdvi xx

או

dvi2tty xx

כאשר המסוף הוא מסוף רגיל הפקודה dvi2tty מציגה את הקובץ.
כאשר עובדים ב-xterminals בחלונות, נשתמש בפקודה xdvi. יש צורך לציין לפני הפקודה xdvi את שם התחנה ע"י הפקודה:

setenv DISPLAY שם תחנה:0

כדי להוציא הדפסה יש להפוך את קובץ ה-dvi לקובץ postscript.
הפקודה לכך היא:

dvi2ps xx.dvi > filep

filep הינו קובץ postscript שאפשר להדפיסו במדפסת לייזר.
כרגע הפקודה מ-CHEN היא:

lpr -Plps filep

DEBUGGER

כדי להשתמש ב-debugger יש לקמפל את התוכנית עם הפרמטר g- , ובהמשך להפעיל את התוכנית עם הפקודה:
dbx קובץ בינארי

יכולים להיות הבדלים קטנים בין מערכת unix אחד לשניה בנושא ה-debugger. הפרטים שיובאו בהמשך קיימים כרגע במערכת AIX. כדי לקבל עזרה בזמן debugger אפשר להקיש:

help

ובכך לקבל את רשימת פקודות ה-dbx.

שלוש דרכים נפוצות להפעלת dbx והם:

1. dbx [אופציות] [קובץ בינארי] [coredump]

אם הקובץ הבינארי לא מצוין, ה-dbx יבקש לקבל את שם הקובץ הבינארי -ברירת המחדל היא a.out .
אם מצויין coredump או קובץ בשם core ב-directory הנוכחי, אזי dbx יציין את המקום בו קרתה טעות ואפשר לקבל ערכים מה-core ולהמשיך בפקודות ה-dbx.

2. dbx - r [אופציות] [קובץ בינארי] [פרמטרים]

r- מציין להכנס ל-dbx רק אם התוכנית נופלת. במידה והתוכנית מסתיימת במילים יש יציאה מה-dbx.
אם יש בעיות בתוכנית תהיה הודעה על הטעויות ונכנסים ל-dbx.

3. dbx [אופציות] -a Process I.D

לבצע debug על תוכנית עובדת ורצה בכדי לדעת את ה-process id אפשר לבצע ממקום אחר את הפקודה:

ps - u loginid

בזמן ה-debug אפשר להריץ פקודות shell. לשם כך יש להקיש קודם:

sh

ובהמשך פקודות shell. כדי לחזור ל-debug יש לכתוב:

exit

<u>נקודות עצירה</u>

ישנם מסי דרכים כדי להציב נקודות עצירה בזמן ה-debug

1. stop at "מציין שורה" [if תנאי]

"מציין שורה" יכול להיות מס' או "מס':שם תוכנית"
"תנאי if" ביטוי שיהיה נכון או לא נכון והוא יקבע אם לעצור.

2. stop in פרוצדורה [if condition]

יעצור בשורת הביצוע הראשונה של הפרוצדורה. לדוגמא:

stop in main

3. stop משתנה [in פרוצדורה | at מציין שורה] [if תנאי]

לעצור כאשר משתנה משנה את ערכו. לדוגמא:

stop x

4. stop if תנאי

יעצור כאשר התנאי מתקיים. למשל:

stop if (x>y) and (x<20000)

ברגע העצירה מתקבל מידע איזה תנאי (איזה stop) גרם לעצירה ומהו ה-event id המיוחס לנקודת עצירה זו.
הפקודה:

status

תציג מידע על המצב הנוכחי.
כדי לבטל עצירה מפעילים את הפקודה:

delete ID

ID הינו ה-identifier event id
כאשר:

delete all

יבטל את כל נקודות העצירה.
כדי לבטל נקודת עצירה לפי מציין שורה בתוכנית, יש להשתמש ב-

clear מסי שורה

הרצת התוכנית

כדי להתחיל להריץ את התוכנית נשתמש בפקודה run

run [פרמטרים] [<קובץ קלט] [>קובץ פלט]

כאשר התוכנית תיעצר עקב נקודת עצירה או סיבה אחרת, תהיה על כך הודעה.

כדי להמשיך משם את התוכנית יש להשתמש בפקודה:

cont

כדי להמשיך את התוכנית בלי debug יש להקיש:

detach

כדי להריץ n צעדים קדימה, נשתמש בפקודה:

step n

לרוץ עד השורה הבאה או מסי שורות ביצוע:

next מסי

ההבדל בין next ל-step הוא כאשר יש קריאה לסברוטינה. אם משתמשים ב-step נעצרים בסברוטינה. ב-next כל הסברוטינה רצה עד שיוצאים ממנו ואז נעצרים. step איטי יותר מ-next.

return - להמשיך את הביצוע עד שנתקלים במשפט return

skip n - תמשיך להתבצע עד סוף התוכנית או עד ש-n+1 נקודות עצירה יתבצעו.

screen - פותח חלון לפקודות קלט ופלט dbx

TRACE

פקודה המציגה מידע בזמן שהתוכנית רצה.

trace in פרוצדורה [if תנאי] - יעצור וידפיס כל שורה בפרוצדורה.

trace procedure [in פרוצדורה] [if תנאי] - מדפיס הודעה כל פעם שנקרא פרוצדורה או חזרו ממנו.

trace ציין שורה [if תנאי] - מדפיס את השורה שמגיעים אליה.

trace ביטוי at ציין שורה [if תנאי] - מדפיס את ערך הביטוי כשמגיעים לשורה.

trace משתנה [in פרוצדורה] [if תנאי] - מדפיס מקום וערך משנה בכל פעם שהמשתנה מקבל ערך שונה.

פקודות נוספות

list - מאפשר לקבל תצוגה של שורות מקור כאשר $ מסמן שורה הבאה לביצוע ו-@ מציין את השורה הבאה להדפסה.

[/] ביטוי / - חיפוש קדימה עבור ביטוי מסויים.

[?] ביטוי ? - חיפוש אחורה עבור ביטוי מסויים.

edit - פקודה המעלה את ה-editor ואז אפשר לעשות שינויים בקובץ.

אפשר להגדיר מה יהיה ה-editor שיעלה, ע"י מתן ערך למשתנה הסביבתי EDITOR לפני שנכנסים ל-dbx וכשיוצאים מה-editor חוזרה ל-dbx.

where - כדי לקבל רשימה של פרוצדורות שנקראו לפני רגע העצירה.

print - כדי להדפיס ערך של ביטוי.

dump - מדפיס את שמות המשתנים והערכים.

assign - כדי לשנות ערך למשתנה.

שם משתנה - שם הקובץ בלי הסיומת נחשב ל-moduls. למשל: משתנה x המוגדר בסברוטינה sub בתוך קובץ hello.c בעלת שם מלא hello.sub.x

. - מציין שם התוכנית.

which - מציג את השם המלא של המשתנה.

whereis - מציג את כל השמות המלאים של המשתנים.

בתוך ביטוי אפשר להשתמש בפעולות הבאות:

פעולות אלגבריות: = , - , * , / , div , mod , exp

פעולות לוגיות: && , || , not , and , or

השוואה: < , > , =< , => , <> , = ! , = או ==

sizeof

(ביטוי) סוג משתנה או סוג משתנה\ביטוי - כדי לשנות סוג של משתנה אפשר להשתמש בפקודה מהפרמטר הנ"ל.

שם whatis - פקודה המציגה את סוג המשתנה.

.dbxinit - מחפשים אותו קודם ב-home directory ואחר כך ב-current directory והוא צריך להכיל פקודות dbx שיבוצעו לפני ה-standart input.

קובץ פקודות - אפשר להכיל קובץ פקודות כלשהוא ולהפעילו ע"י ציון

'קובץ פקודות' dbx - c

EMACS EDITOR

emacs הינו editor המותקן בכל התחנות הקיימות כרגע במרכז החישובים. emacs הינו full screen editor עם מגוון רחב מאוד של פקודות ואפשרויות עבודה. פקודות emacs הם control sequences או esc seguences.
כדי להקל על העבודה הוגדרו מקשי פונקציות המבצעות פקודת emacs. על כן משתמש רגיל לא יצטרך לזכור את הפקודות המורכבות ובכל שלב הוא יכול להציג את הקובץ המכיל את פירוט הפקודות המתבצעות עם כל מקש ומקש.
כדי להפעיל את emacs יש להקיש:

שם קובץ emacs

כתגובה מתקבלים: חלון בו מופיע הקובץ, שורת מידע ברקע כהה בתחתית המסך ושורה אחרונה פנוייה לקלט פלט של פקודות emacs.
emacs עובד על עותק של הקובץ המקורי וזה יתעדכן רק אם נבקש זאת.
כדי לעבוד עם מקשי הפונקציות יש להמצא ב-VT100 mode. mode זה קיים בכל המסופים.
מקשי הפונקציות הם המקשים המופיעים בצד ימין על לוח המקשים והם ה-numerical keypad.

PF1	PF2	PF3	PF4
7	8	9	-
4	5	6	,
1	2	3	
0		.	ENTER

כדי לקבל תצוגה של מה כל מקש מבצע יש להקיש על מקש 0 ובו זמנית על המקש PF4 ומתקבל התצוגה הבאה:

01 First	02 Last	03 GotoLine	04 Copy	05 TermCom	06 Quit
1 Bkw	2 Fwd	3 WhatLine	4 Help	5 Undo	6 SaveQuit
07 DelChar	08 DelToEol	09 ClearRec	OPF1 KillReg	OPF2 InsertFile	
7 DelBaLins	8 Insl	9 OpenRec	PF1 Mark	PF2 ON/OFF overwrite	
OPF3 WardLoc	OPF4 KeysHelp	0- DelOtherWin	0, SkpBol		
PF3 Locate	PF4 LocAll	- OtherWin	, SkpEol		
0Enter DiredOth	O.EnlargeWin				
Enter Refresh	, Askill				

כאשר:

Bkw - מגלגל את הקובץ מסך אחד אחורה.

First - מביא את הסמן לשורה הראשונה בקובץ.

Fwd - מגלגל את הקובץ מסך אחד קדימה.

Last - מביא את הסמן לשורה אחרונה בקובץ.

WhatLine - מציג בשורה האחרונה במסך את מסי השורה בה נמצא הסמן.

GotoLine - הסמן קופץ לשורה האחרונה יש לתקתק את מסי השורה אליה רוצים להעביר את הסמן.

Help - מציג את שורת ה-help של emacs אפשר להקיש כל אחד מהאותיות המוצגות.

copy - לחיצה על מקש זה יעתיק למקום בו נמצא הסמן את תוכן ה-Kill buffer. כדי להעתיק משהו יש צורך קודם לבצע על אותו קטע killאו askill הסבר בהמשך.

Undo - מבטל אחורנית. כל לחיצה על Undo יבטל את הפקודה הקודמת.

TermCom - מקש הגומר פקודה.

Save Quit - מעדכן את הקובץ המקורי ויוצא מה-emacs.

Quit - יציאה מה-emacs בלי לעדכן את הקובץ המקורי.

DelBaLins - מוחק את כל השורות הריקות הקיימות מסביב לסמן פרט לשורה ריקה אחת אם ישנה רק שורה אחת, אזי שורה זו תתבטל.

DelChar - מוחק תו.

Insl - הכנסת התו new line nl במקום המצביע. אם נמצאים באמצע השורה אזי השורה תחצה לשניים. אם נמצאים בתחילת השורה אזי נוצרת שורה ריקה.

DelToEol - מוחק מהמקום בו נמצאים עד סוף השורה.
אם נמצאים בסוף השורה אזי נמחק תו ה-nl והשורה הבאה מתחברת לשורה הנוכחית.
כדי לבטל שורה במלואה, יש להגיע להתחלת השורה ולהפעיל פעמיים את הפונקציה.

Open Rec - Rec עבור Rectangle זהו מלבן שנוצר ע"י ביצוע Mark. לחיצה על מקש PF1 בפינה אחת והעברת הסמן לפינה השניה. הפעלת הפונקציה OpenRec תפתח מלבן זה ותזיז את תוכן הקובץ ימינה.

Clear Rec - כמו Open Rec אך התוכן שהיה במקום נמחק.

Mark - מציב סימן mark באותו מקום שהמצביע נמצא על המסוף לא מופיע שינוי ויזואלי.

KillReg - Reg (Region) מוגדר כאוסף השורות מממיקום ה-mark עד מיקום המצביע והפונקציה הנ"ל מוחקת את הקטע ומצמצם רווחים ומעבירה אותו ל-buffer ממנו אפשר להעתיקו אחר כך

overwrite ON/OFF - emacs עובד כך שכל תו דוחק את התווים הקודמים. במידה ורוצים לדרוך על התוים הקיימים יש להכנס ל-overwrite mode על ידי הפעלת פונקציה זו נעשה ON או OFF למצב ה-overwrite.

Locate - הסמן קופץ לשורה האחרונה שם יש להקיש את הביטוי אותו מבקשים למצוא.
מה-locate אשפר לבצע:

1-חיפוש המופע הבא של הביטוי ע"י	ctrl s
2-חיפוש המופע הקודם של הביטוי ע"י	ctrl r
3-יציאה מהחיפוש וחזרה למקום שממנו התחלנו את החיפוש	ctrl g
4-יציאה מהחיפוש והשארות במקום ע"י מקש	Esc

WordLoc - חיפוש בדומה ל-locate אך הפעם יש למצוא מילה.

LocAll - אחרי הכנסת הביטוי תופיע הודעה בשורה התחתונה המציינת כמה פעמים נמצא הביטוי ובמסך התחתון יוצגו השורות ומספריהם.

KeysHelp - מציג את הקובץ emacs-help. המכיל את טבלת הפונקציות.

OtherWin - מעביר את המצביע מחלון לחלון.

DelOtherWin- מוחק את החלון השני.

SkpEol - קופץ לסוף השורה.

Skpbol - קופץ לתחילת השורה.

Refresh - רושם את המסך מחדש.

DiredOthe - הסמן קופץ לסוף המסך ויש להכניס את ה-directory עליו רוצים לעשות בדיקה. נפתח חלון המכיל את פלט הפקודה ls - al בקובץ זה אפשר להתקדם ולציין פקודות לגבי כל קובץ או כל שורה מוצגת בלי הבדל באיזה מקום בשורה נמצאים.

הכנסת התוים הבאים אפשריים:

d - יציין את התו D בתחילת השורה וזה מסמל שקובץ זה מיועד לזריקה
m - יציין לבטל את הקשת המחיקה
x - יבצע בפועל את מחיקת הקובץ
- מציין שכל הקבצים ששמם מופיע בין '#' מיועדים למחיקה
~ - מציין שכל הקבצים המסתיימים ב-'~' מיועדים למחיקה
c - יעתיק מיד את הקובץ לקובץ שאת שמו נתקתק בסוף המסך
f - עובר לעשות enacs על הקובץ
o - כמו f אבל הקובץ בו נמצא המצביע יוצג בחלון נפרד וחלון ה-dired נשאר
r - לתת שם חדש לקובץ
v - עושה view לקובץ אפשר להסתכל בו אבל לא לשנות אותו

Askill - עובד בדומה ל-killregהאזור המסומן מועתק ל-buffer ממנו אפשר לבצע copy אבל הוא נשאר גם במקום הנוכחי

EnlargeWin - מרחיב את החלון הנוכחי בעוד שורה

פרטים נוספים על emacsבדף פירסום 92/01 במרכז החישובים.

תקשורת בין משתמשים

Who

פקודה המדפיסה מי והיכן המשתמשים עשו login למערכת. הפקודה היא:

who [am I]

הפקודה who בלי פרמטרים תדפיס את הפרטים הבאים: שם משתמש, שם מסוף (ttynames), זמן ה-login של המשתמש.

הפקודה who am I תדפיס את פרטי המשתמש: שם ה-host (המערכת), שם ה-user, שם המסוף, זמן ה-login ושם ה-display של המשתמש.

Finger

מדפיס מידע על משתמש מסויים או על כל המשתמשים במערכת כלשהיא:

finger [options] [user | user @ host | @ host]

הפרטים המוצגים הם: שם ה-user, שם מלא של המשתמש, שם מסוף וסטטוס הכתיבה עליו, idle time, זמן ה-login, מקום המשרד ומסי טלפון של כל אחד ממשתמשי המערכת באותו רגע.

במידה ולא קיימת אפשרות כתיבה על המסוף יופיע ' * ' לפני שם המסוף.

idle time הוא בדקות. אם מופיע ' : ' אזי הזמן הוא בשעות ודקות. אם מופיע ' d ' אזי הזמן הוא בימים ושעות.

אפשר לקבל פרטים גם על מחשב אחר אם כותבים את ה-hostname.

Mesg

פקודה המאפשרת או מבטלת קבלת הודעות messages. הפקודה היא:

mesg [n] [y]

אם הפרמטר n מתבטלת האפשרות שמשתמש אחר יכתוב על מסוף המשתמש ע״י הפקודה write או talk.
אם פרמטר y ניתנת אפשרות כתיבה. בלי שום פרמטר נקבל את המצב העכשווי של אפשרות קבלת הודעות.

Write

כותב הודעה למשתמש אחר. הפקודה היא:

write <u>user</u> [ttyname]

הפקודה מעתיקה שורות ממסוף המשתמש למסוף משתמש אחר בפעם הראשונה המשתמש השני מקבל הודעה

Message from yoursystem! yourname yourttyname

התקשורת נמשכת עד לקבלת סימן end of file (ctrl D) או עד שנעשה interrupt.

מקובל לחכות עד שהמשתמש השני כותב וגומר לכתוב ורק אז להמשיך.
מקובל לסיים את ההודעה בסימן מוסכם לכם שזה הסוף. מקובל ש- ' o ' מסמן 'over' ובכך ההודעה הושלמה ומקובל ש- ' oo ' מסמן 'over & out'.

Talk

הפקודה talk היא תקשורת ויזואלית בין שני משתמשים כאשר שורות מהמסוף מועתקים למסוף של משתמש אחר.
הפקודה היא:

talk <u>person</u> [ttyname]

כאשר person הוא:

- אם מדברים עם משתמש באותה מערכת אזי person הוא חשבון המשתמש במערכת
- אם המשתמש נמצא במערכת אחרת אזי person הוא user @ host

כאשר המשתמש מופיע במערכת מספר פעמים אזי יש לציין את שם המסוף אליו אנו מדברים. לדוגמא user1 יכתוב:

system1 > talk user2 @ system2

המשתמש השני יקבל הודעה מהצורה:

Message from Talk Daeman @ system2 .....
talk: communication rquested .....

על המשתמש השני לכתוב:

talk user1 @ ststem1

ברגע שנוצר הקשר המשתמשים יכולים לכתוב במקביל והפלט מופיע בחלונות נפרדים.
Ctrl L יציג מסך מחדש במידה ויש בו צורך.
כדי לצאת יש להקיש Ctrl C.
הרשאה ל-talk אפשר לבטל או לאשר ע״י הפקודה mesg.

Telnet

interface המאפשר תקשורת עם host מרוחק דרך פרוטוקול Telenet. הפקודה היא:

telnet <u>[host]</u>

כאשר host הוא ה-hostname או כתובת Internet שהיא רצף מספרים המופרדים בנקודה (' . ').

Nslookup

במערכת CHEN אפשר לברר כתובת Internet של מערכת שאת שמה יודעים ע״י הפקודה:

nslookup hostname

נספח 1

טבלת פקודות

פקודה	עמ'	מטרת הפקודה
alias	13	מתן כינוי חדש לפקודה כלשהיא.
cat	12	חיבור קבצים.
cc	17	קימפול תוכנית C במערכות ULTRIX.
cd	8	העברה ל- directory אחר.
chmod	10	שינוי הרשאות לגבי קובץ.
compress	12	דחיסת קובץ.
cp	10	העתקת קובץ.
תו ctrl	6	מספר פקודות עם ctrl.
dbx	19	ריצה עם debugger.
df, du	12	תפוסת נפח דיסקים.
emacs	22	עבודה עם תוכנת העריכה emacs.
fg	15	הבאת job ל-foregrount.
finger	24	לקבלת פירוט על המשתמשים במערכת כלשהיא.
ftp	16	העברת קבצים בין מערכות שונות.
f77	17	קימפול תוכנית fortran במערכת ULTRIX.
grep	18	חיפוש ביטוי בקובץ.
head	11	הצגת תחילת הקובץ על המסך.
history	13	תצוגת מיספור ורישום הפקודות שבוצעו.
jobs	15	מתן מידע על מצב ה-jobs ברגע זה.
kill	15,16	זריקת jobs או תהליך מהמערכת.
logout	3	יציאה מהמערכת.
ls	10	קבלת רשימת קבצים ב-directory.
lpr	13	הדפסת קובץ.
man	6	קבלת online manual של פקודה.
mesg	25	ביטול או אישור קבלת הודעות.
mkdir	9	יצירת directory.
more	11	להציג קובץ על מסך דף דף.
mv	11	העברת קובץ.
nslookup	25	לבירור כתובת internet.
path	14	קביעת מסלול חיפוש פקודה.
pc	17	קימפול תוכנית pascal במערכות ULTRIX.
prompt	3	שינוי prompt.
ps	15	קבלת רשימת התהליכים הרצים כרגע.
pwd	8	קבלת שם ה-directory הנוכחי.
rm	11	מחיקת קובץ.
rmdir	9	מחיקת directory.
source	18	להפעלת קובץ פקודות.
tail	11	להצגת סוף הקובץ.
talk	25	דו שיח בין שני משתמשים.
telent	25	לתקשורת עם מחשב אחר.